JOURNAL OF COMPUTATIONAL PHYSICS **138,** 734–765 (1997)
ARTICLE NO. CP975843

Performance of Under-resolved Two-Dimensional Incompressible Flow Simulations, II[1]

Michael L. Minion* and David L. Brown†

**Mathematics Department, University of North Carolina, Chapel Hill, North Carolina 27599; and †Computing, Information, and Communication Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545*
E-mail: *minion@math.unc.edu and †dlb@lanl.gov

Received December 26, 1996

This paper presents a study of the behavior of several difference approximations for the incompressible Navier–Stokes equations as a function of the computational mesh resolution. In particular, the under-resolved case is considered. The methods considered include a Godunov projection method, a primitive variable ENO method, an upwind vorticity stream-function method, centered difference methods of both a pressure–Poisson and vorticity stream-function formulation, and a pseudospectral method. It is demonstrated that all these methods produce spurious, nonphysical vortices of the type described by Brown and Minion for a Godunov projection method (J. Comput. Phys. **121,** 1995) when the flow is sufficiently under-resolved. The occurrence of these artifacts appears to be due to a nonlinear effect in which the truncation error of the difference method initiates a vortex instability in the computed flow. The implications of this study for adaptive mesh refinement strategies are also discussed.

Key Words: incompressible flow; mesh refinement; numerical accuracy.

1. INTRODUCTION

Even with the most powerful computers available today, the ability to accurately and effectively model realistic engineering problems that include incompressible fluid flow is limited by the memory size and total processor speed of the computer

[1] This work supported by the U.S. Department of Energy through Contracts W-7405-ENG-36, DE-FG02-92ER25139, and DE-AC03-76SF00098.

0021-9991/97 $25.00

being used. An inevitable consequence of this situation is the necessity for a scientist or engineer to be able to make computations with as little resolution as possible while still maintaining sufficient accuracy to obtain reasonable results. For this reason, it is important to understand the behavior of difference methods at the limits of under-resolution. Simply put, an under-resolved computation is one in which the grid spacing of the computational mesh is too large to resolve the smallest physically relevant scales of the solution. For the Navier–Stokes equations the smallest physically relevant scale is determined by the magnitude of the physical viscosity [17]. A surprising aspect of such computations is that the numerical solution may appear to be quite smooth relative to the computational mesh despite being inadequately resolved.

Many modern difference methods are designed to compute solutions that do not exhibit mesh-scale oscillations or become catastrophically unstable regardless of whether the flow is being fully resolved. In addition, such methods attempt to compute small-scale features such as shear layers with as close to the correct scales as can be supported by the computational mesh. We characterize such methods as being *robust.* We contrast these methods both with *nonrobust* methods, which are unable to resolve such features without producing mesh-scale oscillations and possibly instability, and also with *artificial viscosity* methods that add enough diffusion to stabilize the computation on any mesh. Many of the latter methods stabilize the computation at the expense of widening the small-scale features well beyond the correct physical length scales. In this paper, we will only consider methods that we have categorized as ''robust'' and ''nonrobust.''

In an earlier paper [9], a study of the behavior of a Godunov projection method for the time-dependent incompressible Navier–Stokes equations on meshes that provided insufficient resolution was presented. This particular method is of the robust category mentioned above, and in that paper an attempt was made to evaluate the reliability of under-resolved numerical solutions using this method. It was shown that for the model problem of a doubly-periodic shear layer pair, spurious vortices typically appear in under-resolved calculations. While these extra vortices might appear to be physically reasonable, they disappear in mesh-refinement studies, thus justifying their classification as ''spurious.''

The numerical experiments discussed in the earlier paper were restricted to the Godunov projection method, with a brief discussion of similar effects observed in a centered-difference approximation as well as reference to related results in the literature. The present paper extends our previous work by making a more exhaustive study of a variety of difference methods of both upwind and centered-difference type on this shear layers. While this model Rayleigh–Taylor problem certainly is not representative of all possible flows described by the incompressible Navier–Stokes equations, it is one of the classic unstable problems of incompressible flow and is a good model for understanding mixing effects. It is a reasonable requirement for any numerical method that it solve this problem with an appropriate degree of accuracy.

As in the earlier paper, only the Navier–Stokes problem is considered here. The inviscid, or Euler problem, is much more difficult to address since without the viscous length scale, there is no dimensionless number that determines a smallest

physical scale. For computations, then, the smallest scale will be determined by the mesh size and the size of any computational dissipation in the numerical method. In this case, it is impossible to quantify the concept of under-resolution.

In Section 3 a brief description of the numerical methods used in the study is presented. We again consider a Godunov projection method [9, 24, 25] and compare it to an ENO method for the primitive variable formulation of the incompressible Navier–Stokes equations [12, 13] as well as an upwind vorticity stream-function formulation [23]. We also consider two centered-difference approximations, Henshaw's fourth-order method with hyperviscosity stabilization [16] and the centered vorticity stream-function method used in [9]. Finally, we also study a pseudospectral method.

After presenting some convergence results in Section 4, we present calculations that further substantiate the speculation made in [9] that the appearance of spurious vortices in under-resolved computations is not an artifact of any particular difference approximation, but occurs due to under-resolution with any of the methods, of either upwind or centered type. Methods with higher accuracy and which favor centered differencing appear to have more resolution on coarser grids than less accurate methods, but with all of the methods studied, the breakdown in accuracy is manifest in the appearance of these extra vortices.

The test problem that we study consists of a doubly periodic shear to which a sinusoidal perturbation perpendicular to the orientation of the shear layers is made at the lowest wavenumber supported by the computational region. In the absence of any additional perturbations, each of the shear layers rolls up in a single vortex as the flow evolves. If an additional perturbation is made at twice this wavenumber, then two vortices will appear on each layer; the appearance of this solution is quite similar to the under-resolved calculations with the spurious vortices. We therefore speculate that the mechanism for the appearance of the spurious vortices is as follows: because of the quadratic nonlinearity in the Navier–Stokes equations, the truncation error for each of the difference methods is also a quadratic function of higher derivatives of the true solution. If the truncation error becomes too large, it therefore tends to perturb the solution at wavenumbers that are a multiple of the fundamental perturbation wavenumber for the simulation. The manifestation of this will be the initiation and growth of additional vortices in the computed solution. We present some truncation error analysis as well as several numerical examples supporting this conjecture in Section 5.

As we pointed out in the earlier paper [9], one of the most alarming aspects of these results is the fact that without careful mesh-convergence studies, it is difficult to tell the difference between features of the solution that are correct and those that are numerical artifacts. For example, comparing gross statistical measurements such as total energy or enstrophy usually is not sufficient to characterize under-resolution for two runs using a given method, one of which contains spurious vortices, and the other of which does not. (They are, however, useful in characterizing the magnitude of the diffusion in a computation, particularly when verifying that this diffusion mimics the true viscous effects.) Nevertheless, since all methods appear to exhibit the same basic behavior, it is important to be able to understand how to use these methods in real engineering applications. Since sufficient resolution

in regions where physical flow instabilities can occur is the key to accurate computations, it is necessary to be able to detect when and where a flow is not being resolved in a given computation. This issue is of paramount importance in the design of adaptive mesh refinement (AMR) algorithms for incompressible flow [1, 25] in which the computational mesh is locally refined as more resolution is required. This approach is briefly discussed in Section 7, and an example is provided demonstrating that even AMR must be implemented prudently to obtain accurate results.

2. THE NAVIER–STOKES EQUATIONS

2.1. *Primitive Variable Formulation*

There are many formulations of the two-dimensional incompressible Navier–Stokes equations, and several of them are used in this paper. The most common form is the primitive variable formulation, which can be written as

$$U_t + (U \cdot \nabla)U + \nabla p = \nu \Delta U \tag{1}$$

$$\nabla \cdot U = 0, \tag{2}$$

where $U = (u, v)^T = U(x, y, t)$ is the fluid velocity vector with components u and v, the horizontal and vertical velocity, respectively; $p = p(x, y, t)$ is the fluid pressure; ν is the (assumed constant) kinematic viscosity; and subscripts denote partial differentiation. This version (and all other versions) of the equations are considered here in the spatially doubly periodic domain $[0, 1] \times [0, 1]$. For this domain, no boundary conditions are needed other than the requirement that the solutions be doubly periodic.

2.2. *Pressure–Poisson Formulation*

The pressure–Poisson form of the Navier–Stokes equations replaces the divergence constraint (2) with an elliptic equation for the pressure. The pressure equation is derived by taking the divergence of the momentum equation (1) and eliminating terms using the divergence constraint (2):

$$\Delta p + U_x \cdot \nabla u + U_y \cdot \nabla v = 0. \tag{3}$$

Together with the momentum equation (1), this is an equivalent formulation of the incompressible Navier–Stokes equations in a doubly periodic domain.

2.3 *Projection Formulation*

''Projection'' formulations of the incompressible Navier–Stokes equations can be written using the Hodge decomposition theorem, which states that any vector field W can be decomposed into two parts, one of which is divergence-free and the

other solenoidal. Moreover, these components are orthogonal (see, e.g., Chorin [10]),

$$W = U + \nabla\phi, \qquad \nabla \cdot U = 0, \qquad U = \mathbf{P}W, \qquad \nabla\phi = (\mathbf{I} - \mathbf{P})W,$$

where $\mathbf{P}$ is the implied projection operator that decomposes the field. Note that $\mathbf{P}^2 = \mathbf{P}$. Using $\mathbf{P}$, the momentum equation can be written as

$$\begin{aligned} U_t &= \mathbf{P}\{-(U \cdot \nabla)U + \nu\Delta U\} \\ \nabla p &= (\mathbf{I} - \mathbf{P})\{-(U \cdot \nabla)U + \nu\Delta U\}. \end{aligned} \tag{4}$$

In practice, the projection is implemented by solving an elliptic equation for ϕ formed by taking the divergence of the Hodge decomposition

$$\Delta\phi = \nabla \cdot W.$$

Once ϕ is calculated, $\nabla\phi$ is subtracted from the non-divergence-free velocity W to obtain the divergence-free velocity U.

2.4. *Rotational Projection Formulation*

It is also possible to formulate a projection form of the equations in terms of the vorticity $\omega = \nabla \times U$, by noting the identity

$$\omega \times U = (U \cdot \nabla)U - \frac{1}{2}\nabla|U|^2,$$

whence (4) becomes

$$U_t = \mathbf{P}\{-\omega \times U + \nu\Delta U\} \tag{5}$$

$$\nabla\left(p + \frac{1}{2}|U|^2\right) = (\mathbf{I} - \mathbf{P})\{-\omega \times U + \nu\Delta U\}.$$

2.5. *Vorticity Stream-Function Formulation*

The vorticity stream-function formulation uses the vorticity ω and stream function ψ as dependent variables, where the stream function satisfies the Cauchy–Riemann equations,

$$(u, v) = (\psi_y, -\psi_x).$$

In these variables, the incompressible Navier–Stokes equations are given by

$$\omega_t = -(u\omega)_x - (v\omega)_y + \nu\Delta\omega \tag{6}$$

$$\Delta\psi = -\omega. \tag{7}$$

3. NUMERICAL METHODS

Here an overview of the numerical methods used in our study is presented. In each case only a brief description of the details of the method is given since further details are available in the References. Also included is a discussion of the intended advantages of each method as well as inherent disadvantages.

3.1. *The Godunov Projection Method*

In our original study [9], the numerical method used to study the effects of underresolution for incompressible flows was the Godunov projection method described in [5] which is based on the method of Bell *et al.* [4]. In this method the momentum equation (1) is discretized using the second-order Crank–Nicolson type formulation

$$\frac{U^{n+1} - U^n}{\Delta t} + \nabla p^{n+1/2} = -[(U \cdot \nabla)U]^{n+1/2} + \frac{\nu}{2}\Delta(U^n + U^{n+1}), \tag{8}$$

where the term $[(U \cdot \nabla)U]^{n+1/2}$ is calculated explicitly using a second-order Godunov procedure. Equation (8) in turn is approximated in a two-step procedure where an intermediate value U^* is first computed by solving

$$\frac{U^* - U^n}{\Delta t} + \nabla \tilde{p}^{n+1/2} = -[(U \cdot \nabla)U]^{n+1/2} + \frac{\nu}{2}\Delta(U^n + U^*). \tag{9}$$

Here $\nabla \tilde{p}^{n+1/2}$ is an approximation to the time-centered pressure gradient. Once U^* has been computed, values of U^{n+1} and $\nabla p^{n+1/2}$ are computed by using a discrete projection $\mathbf{P}$,

$$U^{n+1} = \mathbf{P}(U^*)$$
$$\nabla p^{n+1/2} = \nabla \tilde{p}^{n+1/2} + (I - \mathbf{P})(U^*)/\Delta t.$$

The Godunov projection method used in the present paper is implemented within an adaptive mesh refinement framework (although for the single-grid problems none of this capability is utilized). There are a few differences between this method and the one used in [9] which originated either as improvements over the former algorithm or as modifications that facilitated the AMR implementation. Some additional details of the AMR method are included in Section 7; a more complete discussion can be found in [25].

One major difference is that an approximate projection operator for which $\mathbf{P}^2 \neq \mathbf{P}$ is used in place of the exact discrete projection in [9]. Approximate projection methods have been introduced in a number of recent papers [2, 19, 25], and they are motivated by the difficulty in implementing an accurate exact projection on locally refined grids as well as difficulties associated with the noncompact nature of discrete ''exact'' projection operators. For a single periodic grid, the approximate projection differs from the projection in [9] only in that the discrete Laplacian used

in the projection Poisson problem uses the standard five-point stencil as opposed to a decoupled five-point stencil.

The second difference concerns the form of the advective derivative terms. The version here follows [24] which presents a modification to the somewhat complicated form of these terms which improves the stability range of the overall method. The specifics will not be presented here.

Finally, the term $\nabla \tilde{p}^{n+1/2}$ appearing in Eq. (9) is computed differently in the current version. In the original, the pressure term was simply lagged, i.e., $\nabla \tilde{p}^{n+1/2} = \nabla p^{n-1/2}$. Here a more accurate value is computed by performing an additional projection. Specifically,

$$\nabla \tilde{p}^{n+1/2} = (I - \mathbf{P})(U^n + \Delta t(-[(U \cdot \nabla)U]^{n+1/2} - \nabla p^{n-1/2} + \nu \Delta U^n)).$$

It should be noted that although these modifications are not trivial, the two versions of the method behave nearly identically on the test problems presented here.

Despite the fact that the method is used here on under-resolved problems with steep gradients in the velocities, no form of slope limiter in the computation of the advective derivatives (as described in the original method) was used, nor was any filtering or smoothing of the flow employed. The combination of the Crank–Nicolson form for viscous terms and the Godunov method for advection is sufficient to make this method robust to under-resolution without the addition of either limiters or filtering.

3.2. *The Pseudospectral Method*

Since the problems considered here are set in a doubly periodic domain, it is possible to solve the equations of motion in spectral space. It is convenient to rewrite the rotational form of the momentum equation (5) as

$$U_t = \mathbf{A}(U) + \nu \Delta U, \tag{10}$$

where

$$\mathbf{A}(U) = \mathbf{P}(-\omega \times U).$$

The Laplacian term is divergence free in spectral space and hence is omitted from the projection.

Denoting the two-dimensional Fourier transform of a function G by $\hat{G}$, it is possible to transform (10) into a system of ordinary differential equations

$$\hat{U}_t = \widehat{\mathbf{A}(U)} - \nu \mathbf{k}^2 \hat{U}, \tag{11}$$

where $\mathbf{k}$ denotes the Fourier wavenumber. The diffusive term is incorporated directly into the ODE through the use of an integrating factor. Define $\Psi = e^{\nu \mathbf{k}^2 t} \hat{U}$ and note that

$$\Psi_t = e^{\nu \mathbf{k}^2 t} \widehat{\mathbf{A}(U)}. \tag{12}$$

The $\widehat{\mathbf{A}(U)}$ term is computed by first forming the vorticity $\hat{\omega}$ in spectral space from $\hat{u}$ and $\hat{v}$. Then the inverse FFT is applied and $\omega \times U$ is computed in real space. Finally, this quantity is transformed back to spectral space where the projection operator is applied. A standard fourth-order Runge–Kutta method is applied to (12) to integrate (11), which after some work gives

$$\hat{U}^1 = e^{-\nu \mathbf{k}^2 \Delta t/2}\left(\hat{U}^n + \frac{\Delta t}{2}\widehat{\mathbf{A}(U^n)}\right)$$

$$\hat{U}^2 = e^{-\nu \mathbf{k}^2 \Delta t/2}\hat{U}^n + \frac{\Delta t}{2}\widehat{\mathbf{A}(U^1)}$$

$$\hat{U}^3 = e^{-\nu \mathbf{k}^2 \Delta t}\hat{U}^n + e^{-\nu \mathbf{k}^2 \Delta t/2}\Delta t\widehat{\mathbf{A}(U^2)}$$

$$\hat{U}^{n+1} = e^{-\nu \mathbf{k}^2 \Delta t}\hat{U}^n + \frac{\Delta t}{6}\left[e^{-\nu \mathbf{k}^2 \Delta t}\widehat{\mathbf{A}(U_n)} + 2e^{-\nu \mathbf{k}^2 \Delta t/2}\left(\widehat{\mathbf{A}(U_1)} + \widehat{\mathbf{A}(U_2)}\right) + \widehat{\mathbf{A}(U_3)}\right].$$

No dealiasing or spectral filtering is done for this method. Consequently, the method is not robust and can become catastrophically unstable when under-resolving flows. Spectral methods have the advantage of being more accurate than other finite difference methods but are not applicable to many engineering problems where complex geometries and/or nonuniform meshes are required.

3.3. *The Pressure–Poisson Method*

This method is adapted from the method for overlapping grids presented by Henshaw [16]. It is based on the pressure–Poisson formulation of the Navier–Stokes equations (1) and (3).

The right-hand side of the momentum equation (1) is discretized using fourth-order centered differences to replace each derivative term. The pressure equation (3) is solved in Fourier space using the spectral representation of the fourth-order centered-difference approximation to the Laplacian. The right-hand side of the pressure equation is also approximated using fourth-order centered differences.

As formulated above, this method contains no artificial diffusion and hence is not robust. The addition of a hyperviscosity term to the numerical method is one way in which the method can be stabilized. For the simulations in Section 5.3, a second-order discretization of $-U_{xxxx} - U_{yyyy}$ scaled by $0.15h^4$ is added explicitly to the physical viscosity terms to damp small scale oscillations in the flow. Note that hyperviscosity is used here only for runs which are badly under-resolved and not in Sections 4 or 5.1.

3.4. *The ENO method*

This method is taken from [12] where it is used in a similar study of the resolution of the method on shear layer flows. It is based on the projection form of the Navier–Stokes equations (4) and uses fourth-order ENO differencing [27, 28] to compute the derivative terms. Viscous terms are computed explicitly using fourth-order centered differences, and the temporal integration is done using a third-order TVD Runge–Kutta method [28].

The projection operator is implemented in spectral space using a slightly different but mathematically equivalent form than that in Section 2.3. Instead of solving a Poisson problem for the gradient piece of the flow, the incompressible velocity is recovered by solving the vorticity stream-function equation (7). Following [12], let $\mathbf{P}(u', v') = (u, v)$. Denoting the Fourier transform of u and v as $\hat{u}$ and $\hat{v}$, the projection in Fourier space is

$$\hat{u} = \frac{d_k(d_k\hat{u}' - d_j\hat{v}')}{d_j^2 + d_k^2}$$

$$\hat{v} = \frac{-d_j(d_k\hat{u}' - d_j\hat{v}')}{d_j^2 + d_k^2}$$

where d_j and d_k are the spectral representations of the fourth-order x and y derivative operator. These are computed by taking the square root of the spectral form of the fourth-order difference approximation to the second derivative, for example,

$$d_k = \sqrt{(-30 + 32\cos(2\pi kh) - 2\cos(4\pi kh))/12h^2}.$$

This somewhat unusual form of the spectral derivative ensures that $\mathbf{P}^2 = \mathbf{P}$.

Before the velocities are transformed back to real space, an 8th-order spectral filter is applied to remove small-scale oscillations from the flow. Each Fourier component is multiplied by

$$\sigma_{j,k} = e^{-\alpha(m(j)^8 + m(k)^8)},$$

where

$$m(i) = \begin{cases} 0 & \text{if } |i| < k_0 \\ \dfrac{|i| - k_0}{N/2 - k_0} & \text{else} \end{cases},$$

and N is the number of terms in the Fourier series. For this method $\alpha = 15\log(10)$, and $k_0 = 4$ [18, 22]. This is a slightly different filter than appears in [12], but was the form implemented in a version of the code kindly furnished by the authors of [12] and subsequently used to verify our implementation. The combination of spectral filtering and the ENO differentiation make this method robust to under-resolution.

3.5. *The Centered Vorticity Stream-Function Method*

This method is the same as was used in [9]. It is based on the vorticity stream-function formulation of the Navier–Stokes equations. The numerical method discretizes the momentum equation (6) by replacing all derivatives on the right-hand side by fourth-order centered differences. The stream function is computed from

the vorticity in Fourier space. Denoting the discrete Fourier coefficients of the vorticity by $\hat{\omega}_{i,j}$, the stream function is given by

$$\psi(x, y) = \frac{1}{4\pi^2} \sum_{j,k=-N/2+1}^{N/2} \frac{\hat{\omega}_{j,k}}{j^2 + k^2} e^{2\pi i(jx+ky)}.$$

The velocities are computed from ψ using fourth-order centered differences, and the temporal integration is done using a standard fourth-order Runge–Kutta method.

The above formulation produces a simple method that is fourth-order accurate in time and space. However, because it contains no artificial diffusion, it is not robust.

3.6. *The Upwind Vorticity Stream-Function Method*

This method is similar in nature to the centered vorticity stream-function method described in the last section, but uses a non-centered differencing for the advective derivatives and is of lower-order accuracy in both space and time.

The specific details of the procedure for calculating the advective derivatives are as follows. (See [23] for a more complete discussion.) The stream function ψ is calculated from the vorticity ω by solving a discretized version of Eq. (7) using the standard five-point discretization of the Laplacian denoted by Δ_5. Specifically,

$$\Delta_5 \psi_{i,j} = \frac{-4\psi_{i,j} + \psi_{i,j+1} + \psi_{i,j-1} + \psi_{i+1,j} + \psi_{i-1,j}}{h^2} = -\omega_{i,j},$$

where $\psi_{i,j} \approx \psi(ih, jh)$ indicate cell-centered values of the approximate stream function. Cell-staggered velocities are then defined from ψ,

$$\begin{aligned} u_{i,j+1/2} &= (\psi_{i,j+1} - \psi_{i,j})/h \\ v_{i+1/2,j} &= -(\psi_{i+1,j} - \psi_{i,j})/h, \end{aligned} \tag{13}$$

and consequently

$$\omega_{i,j} = (v_{i+1/2,j} - v_{i-1/2,j} - u_{i,j+1/2} + u_{i,j-1/2})/h.$$

Using the velocities (13) and terminology from [15], MAC-staggered velocities are defined by

$$u_{i+1/2,j} = (u_{i,j+1/2} + u_{i+1,j+1/2} + u_{i,j-1/2} + u_{i+1,j-1/2})/4 \tag{14}$$

$$v_{i,j+1/2} = (v_{i+1/2,j} + v_{i-1/2,j} + v_{i+1/2,j+1} + v_{i-1/2,j+1})/4. \tag{15}$$

It is easy to see that these velocities have MAC divergence zero, i.e.,

$$\frac{u_{i+1/2,j} - u_{i-1/2,j} + v_{i,j+1/2} - v_{i,j-1/2}}{h} = 0.$$

MAC-staggered values of the vorticity are then interpolated from cell-centered values using an ENO-type procedure to determine the interpolation stencil. Specifically, the first point in the stencil is chosen by upwinding, and the remaining points are chosen by the smoothness of the surrounding points. Once these values are computed, the advective derivatives are calculated conservatively,

$$\mathbf{A}(\omega)_{i,j} = (u_{i+1/2,j}\omega_{i+1/2,j} - u_{i-1/2,j}\omega_{i-1/2,j} + v_{i,j+1/2}\omega_{i,j+1/2} - v_{i,j-1/2}\omega_{i,j-1/2})/h.$$

The viscous term is incorporated explicitly and calculated using Δ_5. Hence the spatially discretized form of (6), (7) is

$$\frac{\partial}{\partial t}\omega_{i,j} = -\mathbf{A}(\omega)_{i,j} + \nu\Delta_5\omega_{i,j}.$$

The temporal integration is performed using the same third-order TVD Runge–Kutta method as in the primitive variable ENO method above.

This method is taken from [23], where it was used to model the evolution of vortex patches and hence possesses two qualities that are desirable for such simulations. First, since the vorticity is differenced conservatively, the method preserves the total circulation of the flow, i.e., the sum of the discrete vorticity does not change in time (it is in fact zero for all the periodic problems studied here). Second, the method is designed to be ''free stream preserving'' which means that regions of constant vorticity will remain constant as they are advected. Also, the non-centered differencing of the advective derivative terms makes this method robust to under-resolution, although the averaging procedure used to compute MAC-staggered velocities in Eq. (14) does add some amount of intrinsic diffusion to the method.

3.7. *Time Step Control*

All of the methods mentioned above use some type of explicit calculation of advective derivatives that necessitates that the time step by limited by a CFL condition. In practice each method recomputes an appropriate time step based on the magnitude of the velocity at each step.

For all but the Godunov projection method the use of a Runge–Kutta temporal integrator allows the time step to be limited by

$$\Delta t < C\frac{h}{\max\limits_{i,j}(|u_{i,j}| + |v_{i,j}|) + 2\nu/h}. \tag{16}$$

In the computations presented, C is given the value 1.8 for the centered vorticity stream-function and pressure–Poisson method. A value of 0.9 is used for both the ENO method and the upwind vorticity stream-function method. (See [28] for stability limits of TVD Runge–Kutta methods.) For the pseudospectral method the same criterion as above is used with $C = 1.8/\pi$ [14] despite the fact that the implicit treatment of the viscous terms can increase the stability region.

For the Godunov projection method, the analysis in [24] suggests that

$$\Delta t < Ch/\max_{i,j}(|u_{i,j}|, |v_{i,j}|) \tag{17}$$

with $C = 1$ is sufficient for stability. Here Eq. (17) is used with $C = 0.9$ as the time-step control.

4. CONVERGENCE STUDIES OF THE NUMERICAL METHODS

In this section results of convergence studies for each of the methods described in the last section are presented. The first of these will demonstrate each of the methods converging to the exact solution of a simple model problem as the mesh spacing is reduced. The second will show convergence of the methods to a highly accurate reference solution of a shear layer problem. Later these results will be contrasted with those obtained from under-resolved simulations.

First a very simple flow is considered for which the exact solution is known. Taking as the domain the doubly-periodic unit square, the following is an exact traveling wave solution of the Navier–Stokes equations

$$u(x, y, t) = 1 + 2\cos(2\pi(x - t))\sin(2\pi(y - t))e^{-8\pi^2\nu t}$$
$$v(x, y, t) = 1 - 2\sin(2\pi(x - t))\cos(2\pi(y - t))e^{-8\pi^2\nu t}$$
$$p(x, y, t) = -(\cos(4\pi(x - t)) + \cos(4\pi(y - t)))e^{-16\pi^2\nu t}.$$

For each of the six numerical methods, the above problem is solved using a $2^n \times 2^n$ computational grid for $n = 4$, 5, 6, and 7. The viscosity for each run was $\nu = 0.01$. For each case the L_1 norm of the error in the u-component of the velocity was computed at time $t = 0.7$, and the results are shown in Fig. 1. The convergence rates quoted in the caption of the figure were computed by taking the $\log_2$ of the ratio of the errors from the 64×64 and 128×128 computations.

A few observations can be made from Fig. 1. The pseudospectral method is by far the most accurate, but this is not surprising since the solution here consists of a single Fourier mode. The fourth-order accuracy of the computed solution is inherited from the accuracy of the time integrator since the spatial accuracy is, in a sense, exact. This is in contrast to the primitive variable ENO method where fourth-order convergence is exhibited despite the method having only third-order accuracy in time. The fact that for this problem the overall error is dominated by the spatial error has nothing to do with the ENO type of discretization; the same behavior is shared by the centered pressure–Poisson method if a less accurate temporal integration method is used. Both of these methods appear slightly less accurate than the centered vorticity stream-function method, a fact probably due to the use of the exact spectral representation of the inverse Laplacian used in the latter. Finally, the two second-order methods produce nearly identical results despite the third-order time integrator of the upwind vorticity stream-function method. Again this is due to the fact that spatial errors dominate the total error, and both have second-order accuracy in space.

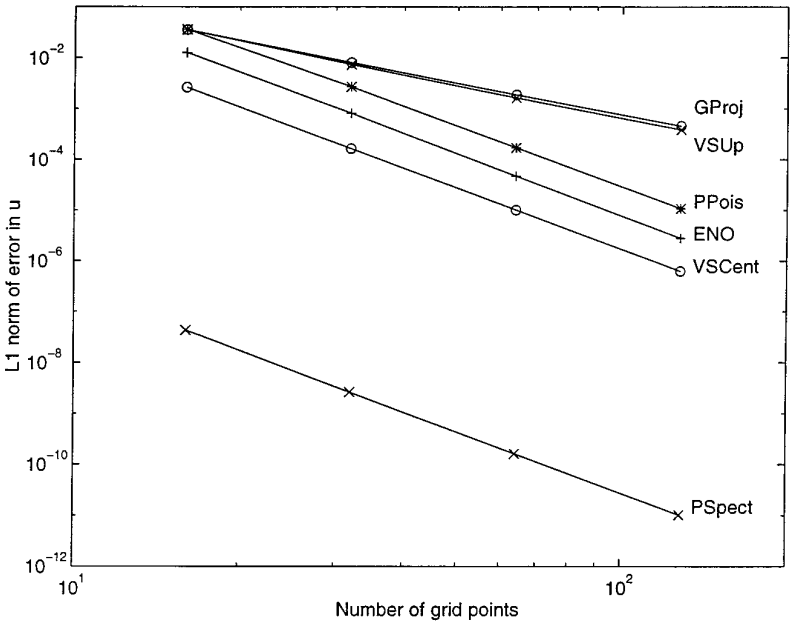

FIG. 1. Convergence results for the traveling wave example. The computed convergence rates are (from top to bottom) Godunov projection, 2.04; upwind vorticity stream-function, 2.08; pressure–Poisson, 3.99; ENO, 4.07; centered vorticity stream-function, 4.01; and pseudospectral, 4.00.

The second convergence test presented uses a problem involving a doubly periodic pair of shear layers. This is the "thick" shear layer problem from [9] with initial conditions given on the periodic unit square

$$u = \begin{cases} \tanh(\rho(y - 0.25)), & \text{for } y \le 0.5 \\ \tanh(\rho(0.75 - y)), & \text{for } y > 0.5 \end{cases} \qquad (18)$$
$$v = \delta \sin(2\pi(x + 0.25)),$$

where ρ is the shear layer width parameter and δ the strength of the initial perturbation. Since an exact solution is not known for this problem, a reference solution computed using a very fine grid was used as the "exact" solution for determining errors in the convergence tests. For the examples to be discussed here, the reference solution was computed to time $t = 1.0$ using the pseudospectral method on a 768×768 grid with $\rho = 30$, $\delta = 0.05$, and the viscosity $\nu = 0.002$. The grid was chosen to be finer than necessary to resolve all physically relevant scales; this can be monitored by examining the spectrum of the flow. Using the highest-order method available assures that the error in the reference solution will be as small as possible.

Each method was used to compute the solution of the thick shear layer problem to time $t = 1.0$ using 64×64, 128×128, and 256×256 grids. The convergence rate of each method was estimated in two ways, by comparison with the reference solution and by a procedure based on Richardson extrapolation.

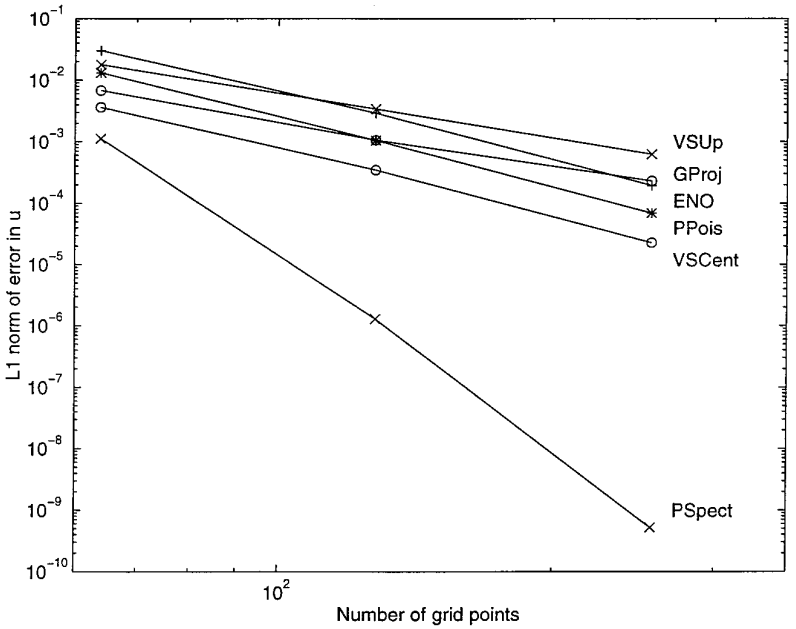

FIG. 2. Convergence results for the thick shear layer example. Shown are results for each method on meshes of size 64×64, 128×128 and 256×256. The computed convergence rates are (from top to bottom) upwind vorticity stream-function, 2.44; Godunov projection, 2.19; ENO, 3.90; pressure–Poisson, 3.92; centered vorticity stream-function, 3.91; and pseudospectral, 11.27.

Figure 2 displays the L_1 errors for each run computed at $t = 1.0$ by comparison with the reference solution. Table I shows the estimated convergence rates for the same runs. The bottom two rows of this table are calculated by comparing computed solutions to the reference solution and correspond to the slope of the line segments in Fig. 2. In the Richardson extrapolation procedure, the L_1 diference between the 64×64 and 128×128 solutions and the L_1 difference between the 128×128 and 256×256 solutions are calculated. These quantities are listed in the first two rows

TABLE I

Estimated Errors and Convergence Rates from the Thick Shear Layer Example. (See Also Fig. 2.) (PSpect = Pseudo-spectral, VSCent = Centered Vorticity Stream-Function, ENO = Primitive Variable ENO, PPois = Pressure-Poisson, VSUp = Upwind Vorticity Stream-Function, GProj = Godunov Projection.)

	PSpect	VSCent	ENO	PPois	VSUp	GProj
64–128 errors	1.05e-3	3.20e-3	1.17e-2	2.60e-2	1.40e-2	5.69e-3
128–256 errors	2.87e-6	3.15e-4	9.61e-4	2.65e-3	2.65e-3	8.35e-4
Richardson rate	8.09	3.35	3.61	3.30	2.35	2.77
64–128 rate	9.75	3.39	3.65	3.38	2.39	2.68
128–256 rate	11.27	3.91	3.90	3.92	2.44	2.19

of Table I and should be a reasonable approximation to the errors in the 64×64 and 128×128 runs, respectively. Taking the $\log_2$ of the ratio of these quantities gives the estimated convergence rate that is listed in the row labeled "Richardson rate." This should be a reasonable estimate to the convergence rate between 64×64 and 128×128 runs, and Table I shows this to be valid. The agreement between the convergence rates estimated for the pseudospectral method using both Richardson extrapolation and comparison with the reference solution further substantiate the claim that it is reasonable to take the 768×768 pseudospectral calculation as a representation of the exact solution. Indeed, the curve for the pseudospectral method in Fig. 2 suggests that the error in the reference solution is substantially less than 10^{-9}.

5. THIN SHEAR LAYERS

To illustrate the performance of the numerical methods on an under-resolved flow, each method is used to compute the evolution of a thin shear layer. The initial conditions are given by Eq. (18) with $\rho = 80$ and $\delta = 0.05$, and the viscosity here is 0.0001. These initial conditions are nearly the same as the thin shear layer problem in [9], but with a slightly thicker shear profile. Note that while we characterize this shear layer as "thin" the transition between states in the shear layer occurs over approximately six mesh points on a 128×128 mesh.

5.1. *Appearance of Spurious Vortices*

The above problem was computed using each method on both a 256×256 and a 128×128 grid. Figures 3 and 4 show vorticity contours from these runs at time $t = 0.6$ and $t = 1.0$, respectively. For each method and time, the plot displays contours from the 256×256 run on the top portion of the plot and contours from the 128×128 on the bottom. (In reality, the 256×256 data is averaged and superimposed onto the coarser grid before plotting.) Since the problem has a top–bottom symmetry, it is sufficient to view only one shear layer from each calculation. The three frames on the right side of each figure show the results for the three robust methods considered while those on the left show results from the three nonrobust methods.

The results from the 256×256 runs appear nearly identical. This is a good indication that all of the methods are capable of computing a reasonably accurate solution, given sufficient resolution. There are, however, substantial differences in the 128×128 runs. Note that the robust methods all produce an additional spurious vortex in the shear layer midway between the main vortices. The nonrobust methods, while they do not produce spurious vortices at this resolution, clearly show the beginnings of unstable oscillations, particularly for $t = 1.0$. As the flow develops further, the latter methods will eventually experience catastrophic blowup as the solution becomes more and more under-resolved.

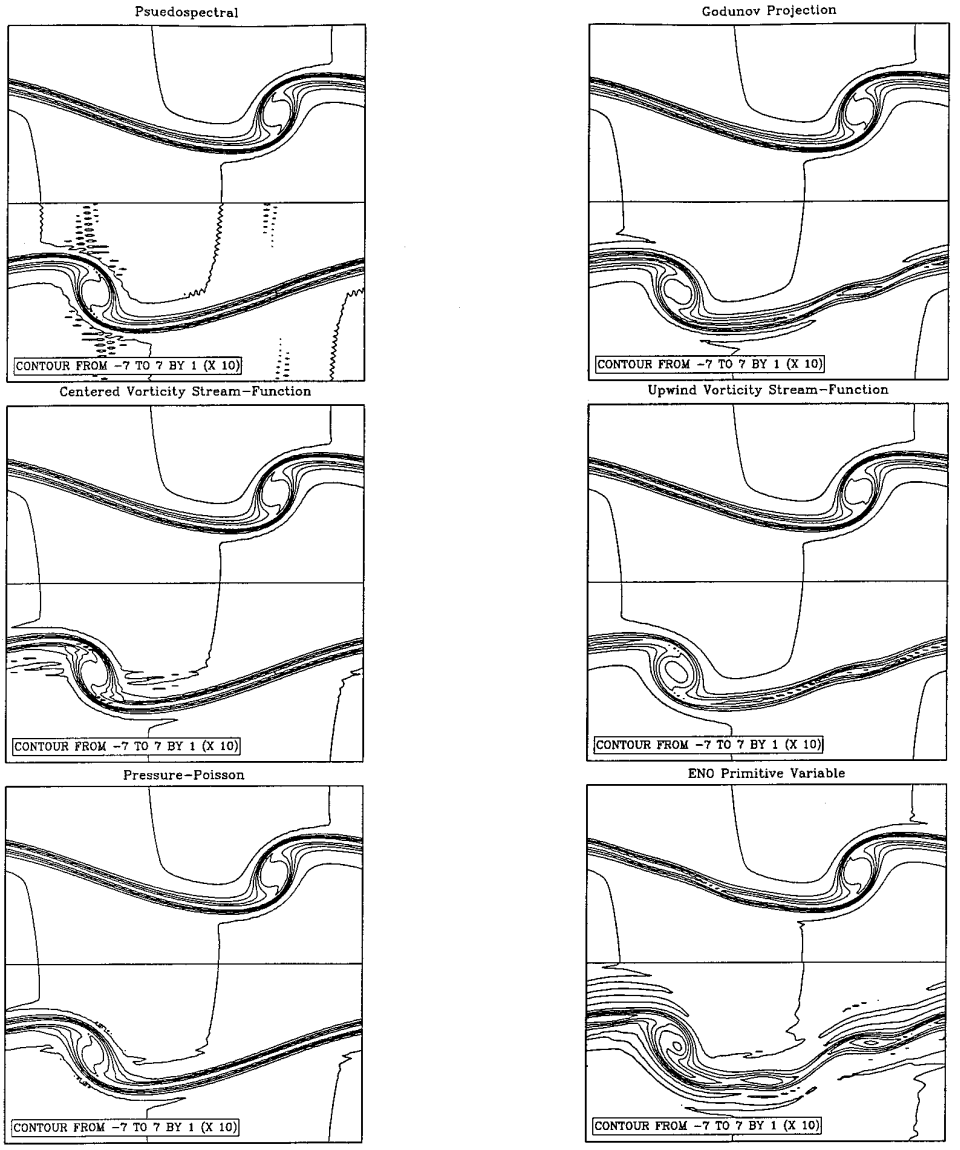

FIG. 3. Contour plots of vorticity from thin shear layer runs at time $t = 0.6$. In each of the six frames, results from the 256×256 computation are shown in the top half of the plot, while results from the 128×128 are shown on the bottom.

5.2. *Convergence Tests*

In Section 4 we demonstrated numerical convergence for each of our methods on a thick shear layer problem. Here it will be shown how these same tests break down in the under-resolved case.

The two convergence study procedures used earlier for the thick shear layer example were repeated using the data from the thin shear layer runs ($\rho = 80$, $\delta = 0.05$, and $\nu = 0.0001$) computed at $t = 1.0$. The results are shown in Table II.

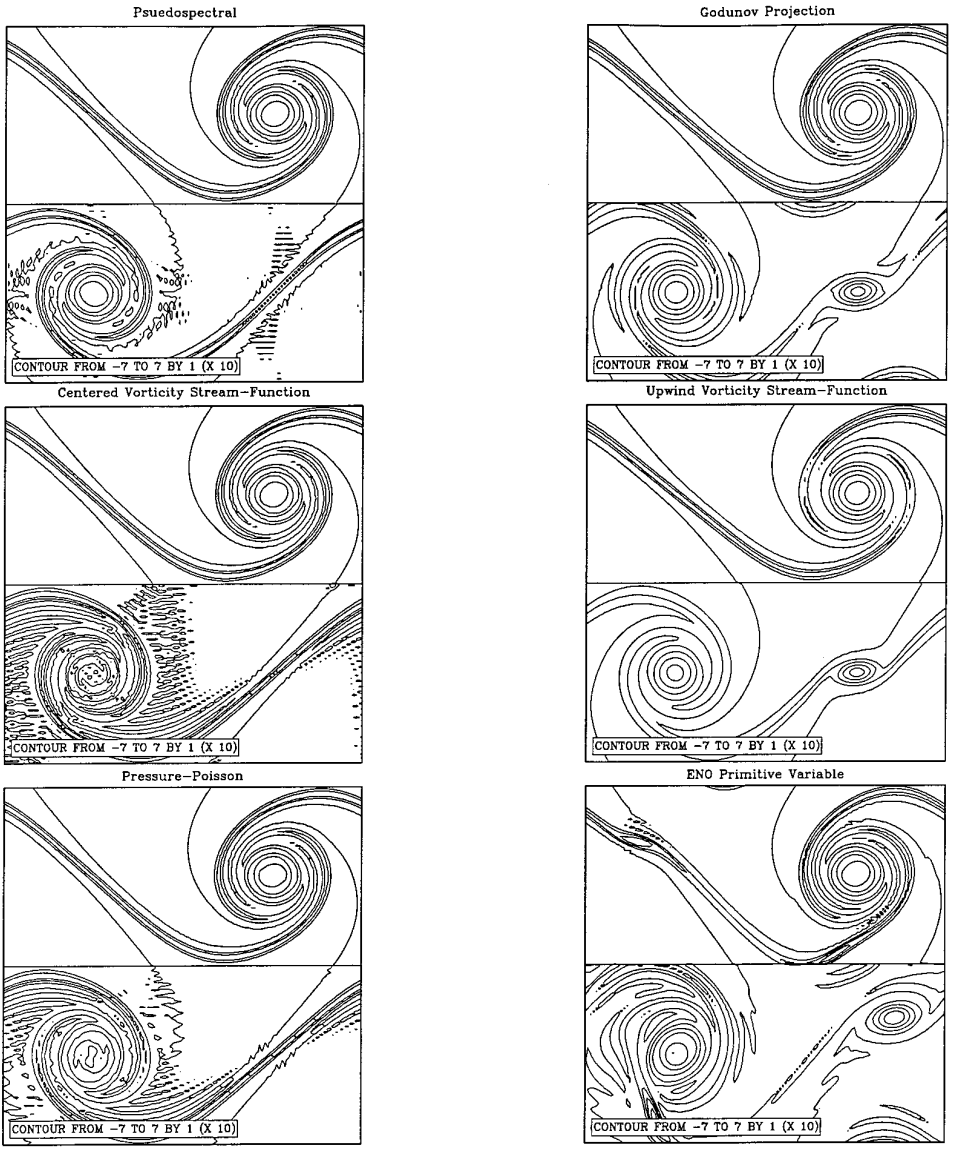

FIG. 4. Contour plots of vorticity from thin shear layer runs at time $t = 1.0$.

Several points should be made. First, it is clear that the convergence rate estimates based on the Richardson procedure no longer indicate that the methods are converging to the proper order of accuracy. These numbers should be reasonable approximations to the convergence rates computed with the 64×64 and 128×128 runs. Here these approximations are not nearly as accurate as in the thick shear layer case, but still give a reasonable indication of how badly convergence is failing. Second, the convergence rates listed in the last row for the upwind vorticity stream-function and Godunov methods are inflated by the fact that the 128×128 runs for these methods contain prominent spurious vortices while the 256×256 runs

TABLE II
Estimated Errors and Convergence Rates from the Thin Shear Layer Example. Labels as in Table I

	PSpect	VSCent	ENO	PPois	VSUp	GProj
64–128 errors	9.47e-2	3.04e-2	8.26e-2	3.79e-2	7.22e-2	9.27e-2
128–256 errors	1.06e-2	3.06e-3	1.05e-1	1.19e-2	6.63e-2	7.45e-2
Richardson rate	3.15	3.31	−0.35	1.66	0.12	0.31
64–128 rate	2.77	1.54	0.48	1.39	0.86	0.97
128–256 rate	11.58	3.63	1.69	3.63	2.53	3.83

do not. Also, hyperviscosity was needed to stabilize the 64×64 run for the pressure–Poisson method which may somewhat affect the convergence rates for this method.

As was pointed out in the [9], it may be difficult to determine from gross statistics whether spurious vortices have appeared in a computation. To demonstrate this, Figs. 5 and 6 display the computed energy and enstrophy of the solutions for the different methods for the thin shear layer problem on a 128×128 grid. The "exact" data, taken from the run used as the reference solution in the convergence tests, are superimposed with open circles on the plot. It is clear that these plots are not

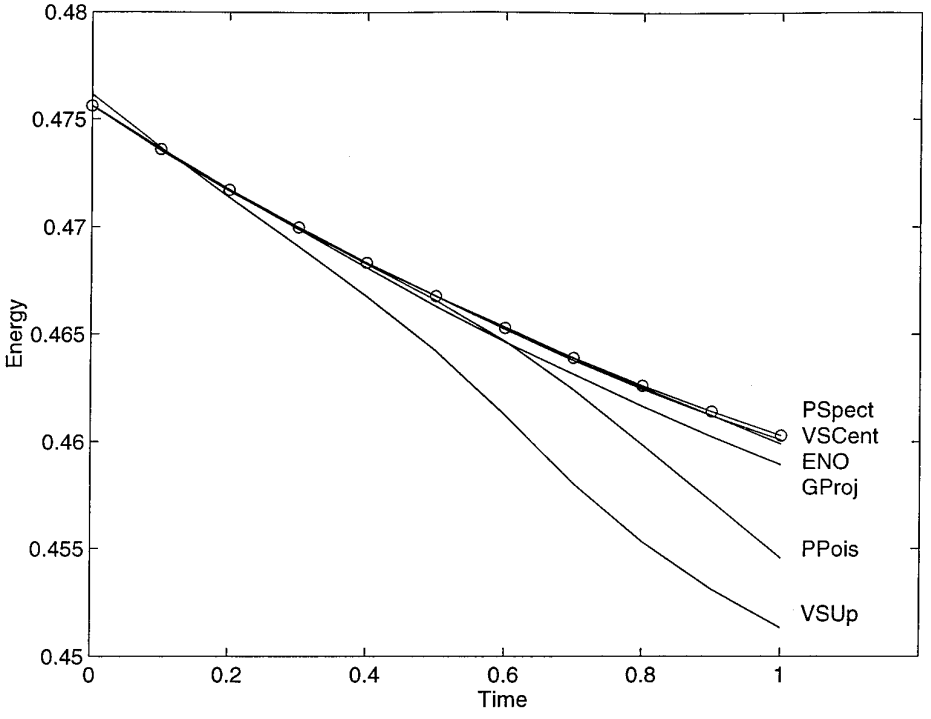

FIG. 5. Energy decay for the thin shear layer example on a 128×128 grid. The data from the "exact" solution are represented by open circles.

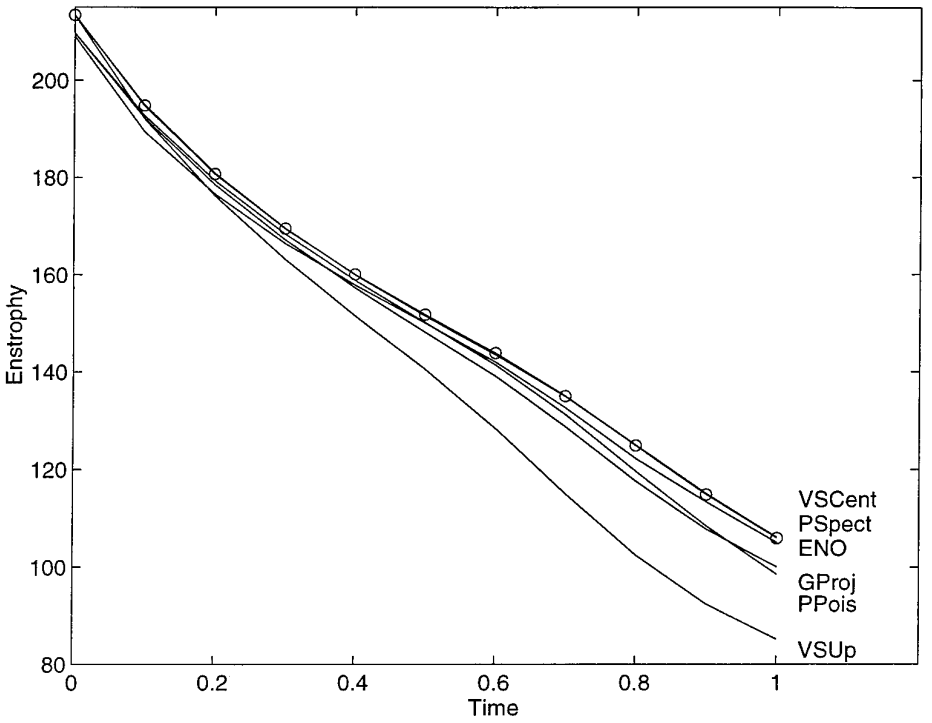

FIG. 6. Enstrophy decay for the thin shear layer example on a 128 × 128 grid. The data from the "exact" solution are represented by open circles.

diagnostic when trying to determine whether physically reasonable solutions are being computed. For example, while the plots in Fig. 4 of the ENO method clearly show the effects of under-resolution, the energy and enstrophy curves in Figs. 5 and 6 for the ENO method lie as close to the exact curves as do those for any of the methods and so would not be a cause for suspicion if viewed independently of any other evidence.

It is also important to note that despite the under-resolved nature of the flows, all of the methods add little numerical diffusion to flow, especially in the early stages of the computation. In other words, it is the *physical* viscosity that is the dominant diffusive mechanism for these runs. A sufficient amount of artificial viscosity will, in fact, prevent spurious vortices (see, e.g., [20]), but at the expense of increasing the width of the underlying shear layers. The mechanism for this is clear; adding too much artificial viscosity can increase the smallest length scale of the problem to the point that it can be accurately represented on the computational grid, suppressing the initiation of such unstable artifacts.

5.3. *Spurious Vortices for the Centered Methods*

The examples of the previous section might give the impression that the nonrobust numerical methods will not produce spurious vortices. However, these methods will also produce incorrect solutions, but on coarser grids or for thinner shear layers

than the robust methods. In this section numerical experiments using very thin shear layers are presented for which this happens.

The pseudospectral, centered vorticity stream-function, and pressure–Poisson methods were run on problems using thinner shear layers on a 128×128 grid. Specifically, for the pseudospectral method, a shear width parameter of $\rho = 100$ and viscosity $\nu = 0.0001$ were used. For the vorticity stream-function method $\rho = 120$ and $\nu = 0.00005$, and for the pressure–Poisson method $\rho = 150$ and $\nu = 0.0001$. Hyperviscosity is used to stabilize the pressure–Poisson method here; hence a thinner shear layer was required than for the other two methods to produce similar results. Contour plots of the vorticity at time $t = 0.6$ for these three runs are shown in the left column of Fig. 7 and at time $t = 1.0$ in Fig. 8. The contour levels for these plots are arranged so that the zero level of vorticity is not plotted, so the substantial amount of small scale oscillation for these runs is not visible.

The left-hand column of these figures clearly shows the appearance of spurious vortices in the shear layers as in the last example, but for the pseudospectral and vorticity stream-function methods, two vortices rather than one appear in each shear layer. Remarkably, if the initial conditions are shifted a half grid cell in the vertical direction, these two methods produce one extra vortex, but the pressure–Poisson method produces two. These runs are shown in the right-hand columns of Figs. 7 and 8. The tendency to produce two versus one spurious vortex appears to be linked to the use of vorticity in the calculation of the advective derivatives. This speculation is supported by the discussion in Section 6.2 and is related to the centering of the initial conditions with respect to the mesh. For the plots in the right-hand column, the zero-line of the vorticity in the initial shear layer lies *along* a horizontal row of discretization points. For the plots in the left-hand column, the zero-line lies *between* two rows of discretization points.

6. SPECULATION ON THE SPURIOUS VORTEX PRODUCTION MECHANISM

It might seem somewhat remarkable that, under conditions of insufficient resolution, all six of the methods presented here break down in the same way, by producing spurious vortices. However, this does suggest that there is an underlying mechanism common to all methods which is responsible for this phenomenon. We discuss such a mechanism in this section.

6.1. *Perturbed Initial Conditions*

For the continuous case, strong shear layers such as those used in the above study are unstable to perturbations in a range of wave numbers. The extent of this range depends on the viscosity of the flow and the thickness of the shear layer (see, e.g., [3, 11]). For the above examples, the shear layer was perturnbed using the lowest available wave number, but higher-frequency perturbations will also cause the shear layers to roll up. Realizing this, we can add a second wavenumber perturbation to the initial conditions above and produce resolved results that look very similar to the spurious vortex results.

Specifically, consider the perturbation flow given by

$$\tilde{u} = \varepsilon \cos(4\pi x + 0.25) \cos(2\pi y)$$
$$\tilde{v} = 2\varepsilon \sin(4\pi x + 0.25) \sin(2\pi y). \tag{19}$$

When added to the basic perturbed shear layer flow (18), this function perturbs the vertical velocity with wavenumber two, i.e., twice the frequency of the original perturbation. Using the value $\varepsilon = 0.00375$, the perturbation in (19) was added to

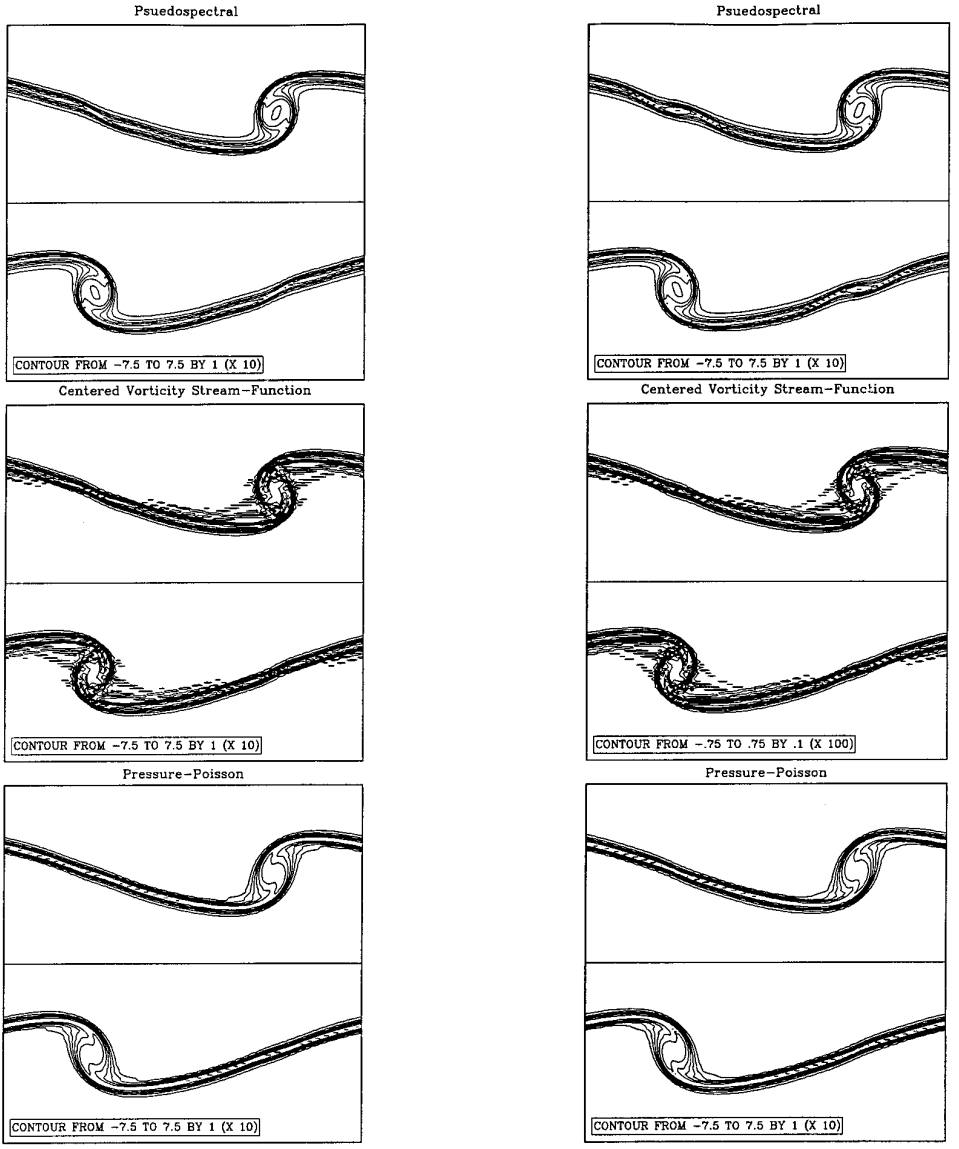

FIG. 7. Contour plots of vorticity from nonrobust methods on very thin shear layers runs at time $t = 0.6$.

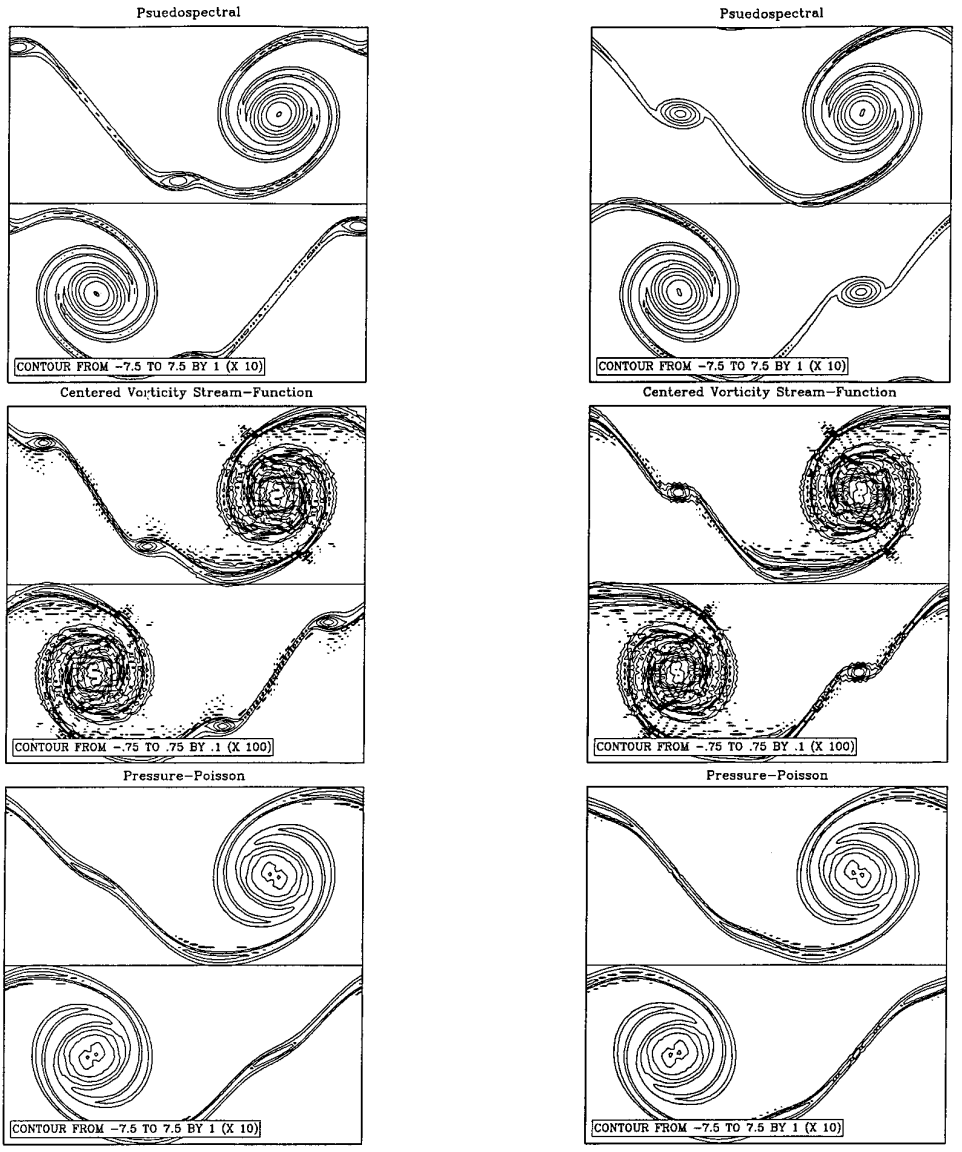

FIG. 8. Contour plots of vorticity from nonrobust methods on very thin shear layers runs at time $t = 1.0$.

the initial conditions (18) using the thin shear layer values $\rho = 80$, $\delta = 0.05$, and $\nu = 0.0001$. The resulting initial value problem was then solved using the centered vorticity stream-function method on a 512×512 grid. The results are shown in Fig. 9. Note that here the extra vortices in the middle of the shear layers are not the result of under-resolution, but are real features of the flow.

This numerical experiment holds the key to understanding the tendency for all of the numerical methods to produce similar results when under-resolving flows. As is well known (see, e.g., [14]), the computational error due to discretization of

a differential equation satisfies the corresponding linearized difference equation forced by the truncation error for the method. By Duhamel's principle, a forced linear problem is equivalent to a superposition of initial value problems for the homogeneous difference equation. Therefore we would expect that if the truncation error contained perturbations similar to (19), then additional vortices might form. For the Navier–Stokes equations, the quadratic nonlinearity in the convective terms will produce a truncation error that is quadratic in derivatives of the exact solution. By appealing to Fourier analysis, we realize that the truncation error will contain Fourier modes at double the frequencies of those that appear in the underlying flow. We therefore speculate that the appearance of the spurious vortices in many of the calculations above is due to truncation error terms that perturb the solution at the second or higher wavenumber. This argument also suggests that in particularly under-resolved calculations, one would expect to see additional spurious vortices at wavenumber four and higher, which in fact was reported in [9] (see, e.g., Fig. 8).

The speculation that terms with wavenumber two in the truncation error of the numerical methods are responsible for the production of spurious vortices is consistent with several key points of our numerical experiments. Most obviously, as the grid is refined, the size of the truncation error decreases, and the spurious vortices in those calculations are less prominent or nonexistent. Also, the fact that it is a low wavenumber feature of the flow that is causing the spurious vortices explains why the use of filtering techniques, such as the exponential spectral filters in the primitive variable ENO method or hyperviscosity in the pressure–Poisson method, does little to affect the formation of spurious vortices. (As we remarked in Section 5.2, however, the use of a sufficiently large amount of artificial viscosity will suppress the spurious vortices, but the width of the shear layer will be increased as well, which in essence defeats the purpose of a ''high-resolution'' method.)

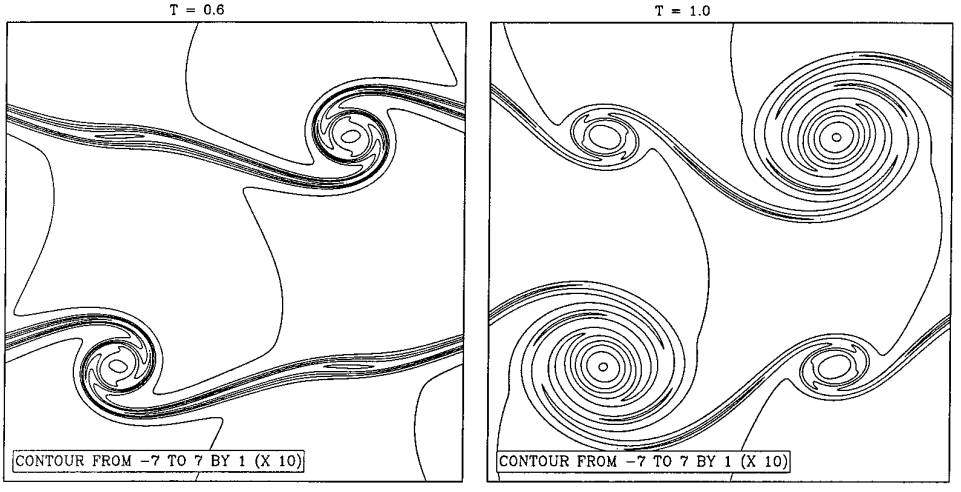

FIG. 9. Highly resolved evolution of perturbed shear layer on a 512 × 512 grid. Note that in this case the additional roll up in the shear layers is not the result of under-resolution.

If our speculation is correct, it should be the case that higher-order accurate methods are less likely to produce spurious vortices than lower-order methods. This is substantiated by the fact that, for instance, reducing the accuracy of the pressure–Poisson method to second order results in a method with a slightly greater tendency to produce spurious vortices. However, we have also observed that the robust methods, even those of higher than second-order accuracy, have a greater tendency to produce spurious vortices than the nonrobust methods. This might be explained partially by the fact that the centering of the numerical differentiation for the two types of methods is different. The robust methods utilize noncentered or upwind differencing near points of under-resolution, while the nonrobust methods are all centered. For methods of the two types that are of the same order, the coefficient of the leading order truncation error terms will be smaller for a centered method than a noncentered one. We have observed however, that a (centered) second-order version of the pressure–Poisson method is still less likely to produce spurious artifacts than is the primitive variable ENO method, which is fourth-order accurate in space. It therefore appears that this truncation error argument does not entirely explain the behavior of these methods.

As an additional comment, note that the form of the time differencing used appears to have little to do with the production of the spurious vortices. As an experiment, the ENO differentiation method in the ENO method was replaced with centered differences of the same order. The result is a method that performs very similarly to the pressure–Poisson method (i.e., resists forming spurious vortices), despite the fact that it is the same order in time as the original ENO method which readily produces spurious vortices. This also suggests that more elaborate ENO methods such as biased or weighted ENO [21, 26] would produce better results than the method used here.

There remains another puzzling result from our experiments, namely, why the pseudospectral method, which is more accurate than any other method tested and is also a centered method, produces spurious vortices so readily. We discuss this in the next section.

6.2. *Truncation Error of the Pseudospectral Method*

The fact that the pseudospectral method so readily produces spurious vortices was initially quite puzzling. After all, it appears to be the most accurate of all the methods considered and should therefore resolve the low wavenumbers of the flow much better than other methods. The explanation for this lies in the formulation of the incompressible Navier–Stokes equations that is used for this method and the particular model problem considered.

Recall from Section 3.2 that the nonlinear terms for the pseudospectral method are based on a projection of the quantity $\omega \times U$. This contains the term uu_y which for the thin shear layer initial conditions is quite unsmooth. In the method, the terms are formed by first computing the vorticity in spectral space, and then forming the quantity $\omega \times U$ in real space. This quantity is then transformed back to spectral space and used in the temporal integration. To examine the aliasing error involved in this procedure, we simply mimic the numerical method and compute the spectral

representation of uu_y for the thin shear layer initial conditions with $\rho = 100$ using only 128 points in the vertical. Figure 10 shows the magnitude in the first 15 wavenumbers of this error. Because of the symmetry of the shear layer, the error first appears in the 4π mode, i.e., the same mode responsible for creating an additional rollup in the shear layer for the other nonspectral methods.

This argument is further substantiated by performing an experiment in which the usual form of the advective terms in the primitive variable formulation $(U \cdot \nabla)U$ is used instead of the rotational form $\omega \times U$. This formulation has the effect of avoiding the aliasing error of the uu_y term. The resulting method is a pseudospectral version of the standard projection method. In this form, instead of a uu_y term appearing in the equation for v, a vu_y term appears in the equation for u. Numerical tests confirm that this formulation can resolve our shear layer problem without producing spurious vortices up to about $\rho = 150$ on a 128×128 grid. This change makes the method very similar to the pressure–Poisson method, and this is evident in numerical experiments, even to the point of producing one spurious vortex as opposed to two and two as opposed to one when the data is shifted a half cell in the vertical as per the discussion in Section 5.3.

The differences in the performance of the two versions of the pseudospectral method seem to disappear if the initial conditions of the problem are rotated with respect to the computational grid. In this case, both partial derivatives of both velocity components are unsmooth. Rotating the initial conditions also removes

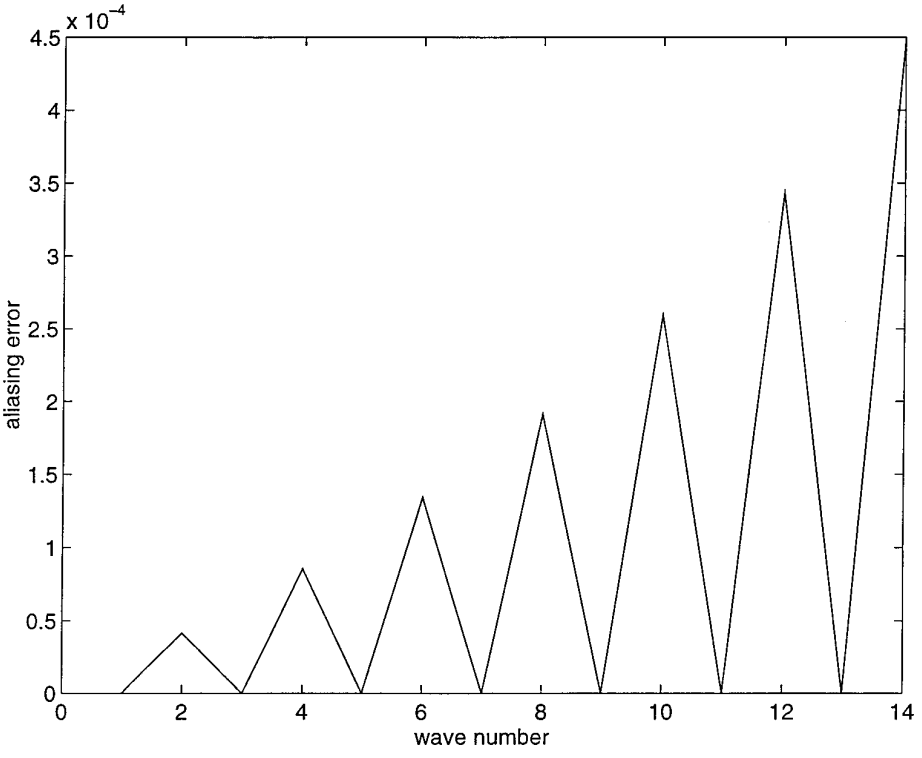

FIG. 10. Aliasing error in the computation of uu_y at $t = 0$ for a 128×128 grid.

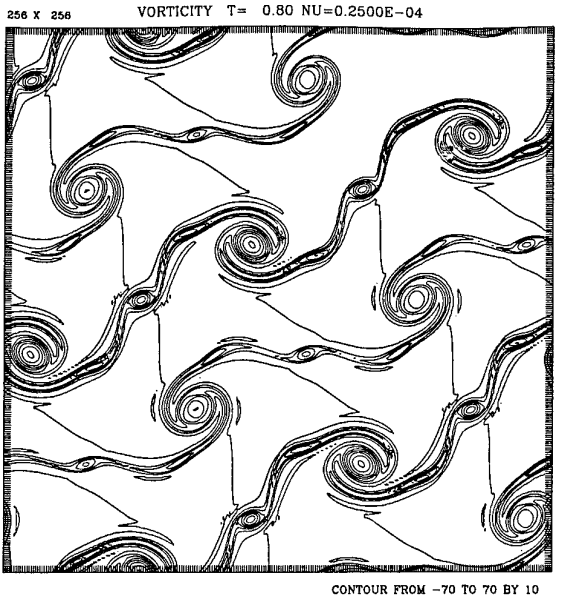

FIG. 11. Contour plots of vorticity with shear layer initial conditions rotated with respect to the computational grid.

any reason to believe that spurious vortices are only produced when the initial shear layers are parallel to the grid axis. Figure 11 shows results obtained with the Godunov projection method using shear layer initial conditions which have been rotated by an angle equal to arctan(1/2). A single spurious vortex is clearly evident in each of the shear layers, and in fact double spurious vortices can be produced by shifting the initial conditions. Similar results can be produced with the other methods as well.

7. ADAPTIVE MESH REFINEMENT

In the design and implementation of numerical methods that use adaptive mesh refinement techniques it is necessary to devise criteria for determining when and where the computational mesh should be refined. In general, some sort of error estimation is used, and the mesh is refined in regions where the error is unacceptably large. Both Richard-extrapolation-based procedures and strategies based on monitoring the local magnitude of flow quantities are common and are discussed in the literature [6–8]. In the context of the present paper, error estimation is essentially equivalent to deciding when and where a numerical method is under-resolving the flow. Additionally, as a time-dependent computation proceeds, grid refinement must occur before nonphysical effects such as spurious vortices appear in the flow. There is therefore an intimate connection between studying the effects of under-resolution and developing effective strategies for AMR. In this section some simple numerical experiments are presented that are designed to emphasize some of the key concerns relevant to using AMR for incompressible flow.

7.1. *A Projection Method for Locally Refined Grids*

The numerical method used for the AMR examples to follow is taken from the projection method for locally refined grids presented in [25]. This method uses a block structured grid refinement strategy similar to that originally described by Berger and Oliger [7, 8]. Within this grid hierarchy, the Godunov projection method described in Section 3 is implemented. Each grid patch is updated every time step regardless of cell size using boundary conditions which are either interpolated from coarser grids or given by the physical boundary conditions (here periodicity). This is in contrast to other methods in which temporal as well as spatial refinement is used [1, 29]. This organization has the advantage that for each of the explicit operations performed in the method, the AMR method for a single grid patch uses the same procedures as a single-grid method. The exception to this design philosophy comes when the implicit parts of the method are implemented.

The Poisson equation associated with the projection step in this method is global in nature and is solved on the entire AMR structure simultaneously. The multigrid technique used, wherein relaxation on individual grid patches is combined with coarsening and interpolation sweeps through the AMR structure, is described in [25]. The method presented in [25] is for Euler flow only and has no description of the solution of the elliptic equation associated with the Crank–Nicolson treatment of the viscous terms. It is, however, straightforward to modify the multigrid method for the Poisson equation to solve this slightly different elliptic equation, and this is what is done here.

7.2. *Local Refinement and Spurious Vortices*

In this section, three calculations performed with the Godunov projection method are presented in which AMR is used in an attempt to resolve the perturbed shear layer flow problem and which illustrate the importance of a careful and effective error estimation technique. In all three examples, an area of the computational grid near the point where spurious vortices are known to occur is refined by a factor of two over the base grid. The refinement is present at the initial time $t = 0$, and the refined regions do not change during the calculations. The difference in the three experiments is the extent of the refined region and the resolution of the base grid.

In the first two examples, a square refinement patch with sides of length 0.25 is centered over the point where a spurious vortex is known to appear. The first example uses a 128×128 base grid so that the effective resolution in the refined patch is that of a 256×256 grid. The computed solution at time $t = 0.8$ is shown in Fig. 12. The location of the refinement patch is indicated by outlining it on the contour plot.

For the same method, a 256×256 grid is known to not produce a spurious vortex, and so since the refinement patch has the same local resolution as that grid, one might expect the resulting computation not to produce such an artifact. However, as is apparent in the figure, this simple attempt at mesh refinement is unsuccessful in suppressing the spurious vortex. The source of the problem has to do with the fact that a factor-of-two refinement occurs in the middle of a region

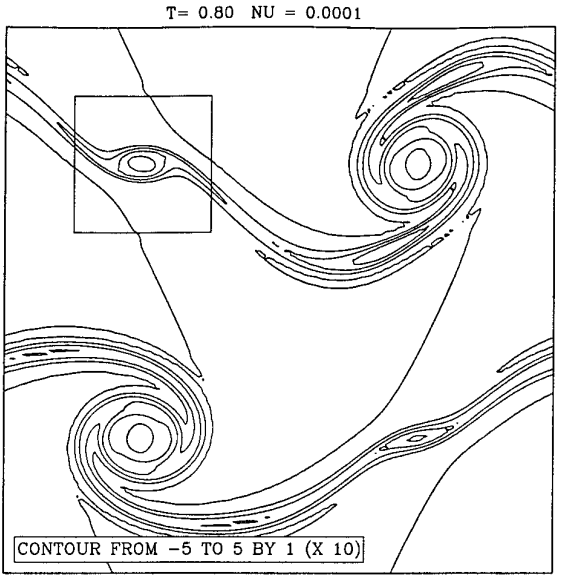

FIG. 12. Contour plots of vorticity with naive mesh refinement.

(i.e., the region around the shear layer) that is not sufficiently resolved on the 128×128 mesh. As a result the shear layer is not represented smoothly across the interface of the coarse and fine meshes, and the spurious vortex is initiated. The same experiment is repeated using a highly resolved 512×512 base grid (giving an effective resolution of 1024×1024 grid), and the results are displayed in Fig. 13. Here the naive grid refinement at least does no damage to the solution up to the end of the calculation at time $t = 1.0$.

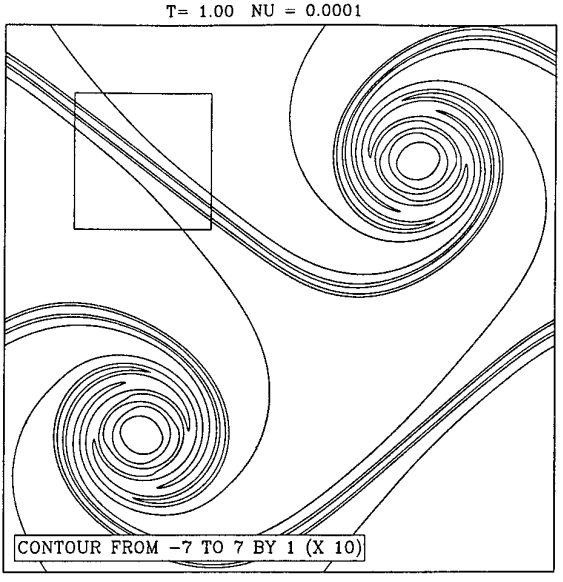

FIG. 13. Contour plots of vorticity with naive mesh refinement but with a resolved coarse grid solution.

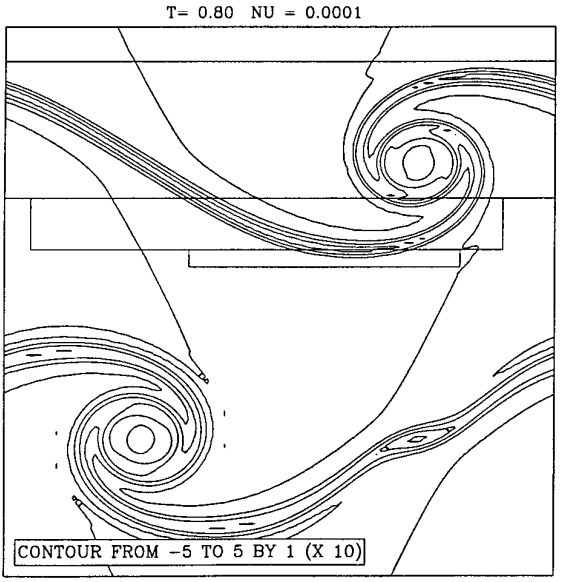

FIG. 14. Contour plots of vorticity using AMR with the entire upper shear layer refined.

Figure 14 shows results from a computation in which the refinement region is initially arranged so that the entire upper shear layer will be contained in the refined region throughout the calculation. A 128×128 base grid is used giving the effective resolution of a 256×256 grid for the upper shear layer. Comparing the upper and lower shear layers, it is seen that the use of grid refinement has prevented the spurious vortex from appearing in the upper shear layer without affecting the lower shear layer.

The lesson learned from this simple comparison is that an AMR strategy for local refinement must be done intelligently. Grid interfaces between coarse and fine grids should not lie across unstable structures unless these structures are fully resolved on the coarse grid. Also, if refinement is to be increased as a computation evolves and smaller scales develop, grid refinement must be employed before large scale effects of under-resolution (e.g., spurious vortices) have occurred. Any procedure for automatically determining where a computational grid should be refined must address these issues.

8. DISCUSSION

In this paper, we have extended our earlier work on under-resolution in incompressible flow calculations [9] by presenting further numerical experiments using a set of upwind and centered algorithms that is representative of many of the currently popular finite difference algorithms for approximating the incompressible Navier–Stokes equations. It has been demonstrated that the production of spurious vortices in the double shear layer problem is a result of insufficient resolution and occurs with both upwind and centered type methods. These artifacts are not a high-wavenumber effect, but rather represent the growth of unstable low-wavenumber

perturbations introduced by the truncation error of the methods. Since these spurious vortices appear as large smooth structures in a computation, it is difficult to differentiate them from the physical features of the flow. It may not be possible to both prevent these spurious vortices by using artificial viscosity or filtering techniques and to maintain the correct physical smallest length scales in a computation. Adding a sufficient amount of artificial viscosity will, in fact, prevent these artifacts, but at the expense of increasing the width of the underlying shear layers. This subverts our stated objective, which is to compute solutions to the Navier–Stokes equations in which small length-scale features are represented with as close to the correct scales as can be supported by the computational mesh.

Spurious vortices can be prevented for all methods by computing on a grid that is sufficiently fine to resolve the smallest scales in the problem of interest. The resolution required clearly depends on the difference method. For example, we have observed that in general centered methods require fewer grid points than do upwind methods in order to prevent these artifacts. It is, not surprisingly, also the case that higher-order accurate methods of a given class require less resolution than do lower-order methods. While this might seem to constitute an endorsement of centered methods, one must keep in mind that in general, it is difficult to stabilize centered methods in the nearly under-resolved case using artificial viscosity, without at the same time swamping the physical viscosity, and hence destroying the desired smallest length-scale resolution.

Choosing a mesh size to optimally resolve the smallest length scales in a problem usually requires a careful mesh convergence study. As we have shown, in the absence of spurious vortices, all of the methods considered here converge at the expected rates as the mesh is refined. As a general strategy, making a mesh refinement study for every computation may not be practical. A more efficient alternative is to use adaptive mesh refinement to increase resolution only where necessary. We have given brief evidence in this paper that a careful AMR strategy will prevent spurious vortices. Using a procedure based on Richardson extrapolation for error estimation may be an effective strategy for determining where to refine the mesh, although monitoring the local magnitude of flow quantities such as the vorticity may also be sufficient. The question of how to implement effective error estimation criteria for this problem is a topic for future research.

ACKNOWLEDGMENTS

The authors thank Weinan E and Chi-Wang Shu for providing us with a version of the ENO method used in this work and Michael Shelley for providing a prototype of the pseudospectral code and many helpful comments. We also thank Ricardo Cortez and Bill Henshaw for helpful advice and careful proofreading.

REFERENCES

1. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier–Stokes Equations,* Preprint LBNL-39075/UC-405, Lawrence Berkeley National Laboratory, July 1996. [Submitted for publication.]

2. A. S. Almgren, J. B. Bell, and W. G. Szymczak, *A Numerical Method for the Incompressible Navier–Stokes Equations Based on an Approximate Projection,* LLNL-unclassified report UCRL-JC-112842, Lawrence Livermore National Laboratory, 1993.
3. G. R. Baker and M. J. Shelley, On the connection between thin vortex layers and vortex sheets, *J. Fluid Mech.* **215,** 161 (1990).
4. J. B. Bell, P. Colella, and H. M. Glaz, A second order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **85,** 257 (1989).
5. J. B. Bell, P. Colella, and L. H. Howell, An efficient second-order projection method for viscous incompressible flow, in Proceedings of the Tenth AIAA Computational Fluid Dynamics Conference, AIAA, June 1991, pp. 360–367.
6. M. J. Berger, M. Aftosmis, and J. Melton, Accuracy, adaptive methods and complex geometry, in *Proceedings of the First AFOSR Conference on Dynamic Motion CFD, June 1996.*
7. M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* **82,** 64 (1989).
8. M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* **53,** 484 (1984).
9. D. L. Brown and M. Minion, Performance of underresolved two-dimensional incompressible flow simulations, *J. Comput. Phys.* **121** (1995).
10. A. J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comp.* **22,** 742 (1968).
11. P. G. Drazin and W. H. Reid, *Hydrodynamic Stability* (Cambridge Univ. Press, Cambridge, UK, 1981).
12. W. E. Shu and C.-W. Shu, *A Numerical Resolution Study of High Order Essentially Non-oscillatory Schemes Applied to Incompressible Flow,* ICASE Rept. 92-39, ICASE, 1992.
13. W. E. Shu and C.-W. Shu, A numerical resolution study of high order essentially non-oscillatory schemes applied to incompressible flow, *J. Comput. Phys.* **110,** 39 (1994).
14. B. Gustafsson, H.-O. Kreiss, and J. Oliger, *Time Dependent Problems and Difference Methods* (Wiley, New York, 1995).
15. F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces, *Phys. Fluids* **8** (1965).
16. W. D. Henshaw, A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids, *J. Comput. Phys.* **113,** 13 (1994).
17. W. D. Henshaw, H.-O. Kreiss, and L. Reyna, On the smallest scale for the incompressible Navier–Stokes equations, *Theoret. Comput. Fluid Dynam.* **1,** 65 (1989).
18. D. A. Kopriva, A practical assessment of spectral accuracy for hyperbolic problems with discontinuities, *J. Sci. Comput.* **2,** 249 (1987).
19. M. F. Lai, J. Bell, and P. Colella, A projection method for combustion in the zero Mach number limit, in *Proceedings of the Eleventh AIAA Computational Fluid Dynamics Conference, AIAA, June 1993,* p. 776.
20. D. Levy and E. Tadmor, *Non-oscillatory Central Schemes for the Incompressible Navier–Stokes Equations,* Tech. Rep. 96-37, Department of Mathematics, UCLA, 1996.
21. X.-D. Liu, S. Osher, and T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* **115,** 200 (1994).
22. A. Majda, J. McDonough, and S. Osher, The Fourier method for nonsmooth initial data, *Math. Comput.* **77,** 439 (1988).
23. M. L. Minion, *Two Methods for the Study of Vortex Patch Evolution on Locally Refined Grids,* Ph.D. thesis, University of California, Berkeley, May 1994.
24. M. L. Minion, A note on the stability of Godunov-projection methods, *J. Comput. Phys.* **123** (1996).
25. M. L. Minion, A projection method for locally refined grids, *J. Comput. Phys.* **127** (1996).
26. G. Shan Jiang and C. Wang Shu, Efficient implementation of weighted eno schemes, *J. Comput. Phys.* **126,** 202 (1996).

27. C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* **77,** 439 (1988).
28. C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, *J. Comput. Phys.* **83,** 32 (1989).
29. D. E. Stevens, *An Adaptive Multilevel Method for Boundary Layer Meteorology,* Ph.D. thesis, University of Washington, 1994.